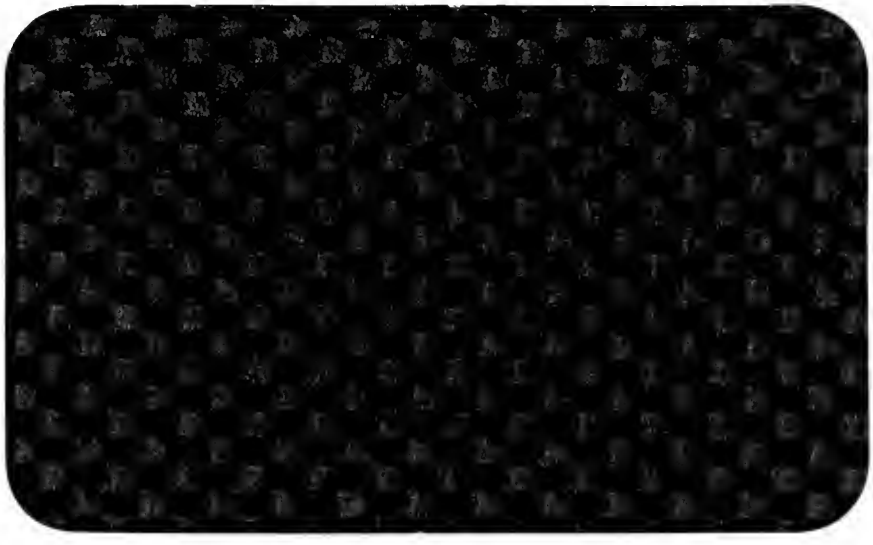
A Domain Decomposition Laplace Solver For Internal Combustion Engine Modeling

Christoph Börgers †
Olof B Widlund ‡

Technical Report #315

August 1987



A Domain Decomposition Laplace Solver For Internal Combustion Engine Modeling

Christoph Börgers †
Olof B. Widlund ‡

Technical Report #315

August 1987

† Supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy, under contract DE-AC03-76SF00098, at the Lawrence Berkeley Laboratory.

‡ Supported in part by the National Science Foundation under grant NSF-DCR-8405506 and by the U.S. Department of Energy under contract DE-AC02-76ER0377-V at the Courant Mathematics and Computing Laboratory.

A Domain Decomposition Laplace Solver for Internal Combustion Engine Modeling

Christoph Börgers [+]

Department of Mathematics

University of Michigan

Ann Arbor, MI 4 9

and

Olof B. Widlund [++]

Courant Institute of Mathematical Sciences

New York University

New York, NY 10012

Keywords: Laplace solver, domain decomposition, multigrid methods

AMS subject classification: 65N20, 65F10

Abbreviation of title: Domain Decomposition Laplace Solver

[+] Supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy, under contract DE-AC03-76SF00098, at the Lawrence Berkeley Laboratory.

[++] Supported in part by the National Science Foundation under grant NSF-DCR-8405506 and by the U.S. Department of Energy under contract DE-AC02-76ER03077-V at the Courant Mathematics and Computing Laboratory.

Abstract: Discrete elliptic problems can often naturally be partitioned into subproblems corresponding to subregions into which the region has been partitioned or from which it was originally assembled. Fast conjugate gradient methods have been developed to handle the continuity conditions, across the interfaces, that need to be satisfied by the solution of the entire problem.

In this paper, we focus on a special family of problems that arises in combustion engine modeling. We demonstrate that efficient algorithms can conveniently be built using standard software to solve elliptic problems on simple regions. We give a convergence analysis which is much simpler than the general theory for domain decomposition algorithms. We also show that the use of multigrid cycles to solve the subproblems inexactly can be very efficient.

A Domain Decomposition Laplace Solver for Internal Combustion Engine Modeling

1. Introduction

The interest in domain decomposition methods has increased considerably in recent years, to a large extent because of their promise on parallel computers. There are frequent conferences; cf. Glowinski and Périaux (1987). Much has been learned on how to design optimal or almost optimal iterative algorithms to handle the interaction between solvers on the subregions. In this paper, we will examine the use of one of these methods for a particular problem arising in combustion engine modeling. We note, however, that the ideas explored here can be adopted to other classes of problems in fluid dynamics etc.. By focusing on a quite specific problem, we are able to demonstrate that the theory sometimes can be simplified, that the algorithms can be speeded up by using special features, and that standard software can be used to assemble a solver on relatively simple but non-trivial regions.

We describe a Laplace solver on a domain Ω of the shape shown in Fig. 1. This solver will be used to compute the potential flow component in a random vortex method applied to a problem with a moving piston; see Sethian (1987). Related work is described, e.g., in Sethian (1984), Sethian and Ghoniem (1986).

In Fig. 1, the large rectangle $\Omega^{(2)}$ represents a cylinder, and the small attached quadrilaterals, the union of which we denote by $\Omega^{(1)}$, represent so-called inlets. There are no inlets attached to the lower side of $\Omega^{(2)}$. The number of inlets attached to the left, right and upper sides of $\Omega^{(2)}$, their positions, and the angles which they form with $\Omega^{(2)}$ are left variable in our program. Sethian's calculations are time dependent with the lower side of $\Omega^{(2)}$ moving. It is a fractional step method, and one of the steps requires the solution of a Neumann problem of the form

$$-\Delta\phi = 0 \text{ on } \Omega \tag{1.1}$$

$$\frac{\partial\phi}{\partial n} = g \text{ on } \partial\Omega. \tag{1.2}$$

Here ϕ is a velocity potential. The Neumann data g and the position of the lower side of $\Omega^{(2)}$, representing the top of the piston, vary with time. The lowest inlet always remains above the lower

side of $\Omega^{(2)}$.

Equations (1.1) and (1.2) are discretized with piecewise linear finite elements on a quasi-uniform mesh as described in Section 2. The region has re-entrant corners, and this is known to limit the regularity of the solution, cf. Grisvard (1985). As a consequence, the discretization will yield only a crude approximation to ϕ, unless the mesh width is very small. On the other hand, the resulting system of linear equations has a relatively simple structure and can be solved very efficiently. We will focus on this case in this paper, but discuss possible remedies through local refinement of the mesh in the last section. One of the domain decomposition methods studied by Bjørstad and Widlund (1984,1986), the so-called "excellent" or Neumann- Dirichlet algorithm, will be used. A careful motivation of this choice is given in those papers. For the problem at hand, we use a fast Poisson solver for the rectangular region and band Cholesky for the relatively narrow inlets. The general convergence analysis of this method requires tools from the theory of elliptic boundary value problems; see Bjørstad and Widlund (1986), Widlund (1987b). However, for the geometry and meshes under consideration, a much simpler convergence analysis, using elementary mappings and reflections, can be given. This results in quantative and realistic bounds; see Section 4, where results of numerical experiments are also presented.

Recently there has been an increased interest in the use of inexact solvers for the problems defined on the subregions, see Bramble, Pasciak and Schatz (1986) and Widlund (1987a). In the fifth section, we demonstrate that the exact solver for $\Omega^{(2)}$ can be replaced by a multigrid cycle. The number of outer iterations grows modestly, but the cost per step is considerably decreased. We also give bounds for the rate of convergence of this variant of the algorithm. In the final section, we collect a number of ideas that could considerably further enhance the performance of the iterative method as well as the accuracy of the discretization..

The code used in the numerical experiments is available from the authors.

2. Triangulation of the region

The region is the union of a rectangle $\Omega^{(2)}$ and N_{inlets} small quadrilaterals called inlets. $\Omega^{(2)}$ is covered by a regular, rectangular mesh, with mesh widths h_x in the x-direction and h_y in the y-direction. We use logically rectangular meshes on the inlets, see Fig. 2, with common mesh points on the interface between any inlet and the rectangle.

The triangulation of the rectangle is obtained by cutting each cell of the rectangular mesh into two triangles. We do not need to specify the direction of these cuts, since the same five point formula results when we use piecewise linear finite elements.

We triangulate an inlet by defining a bilinear mapping ψ_i from a rectangle R_i onto the inlet, I_i, as indicated in Fig. 2. On the rectangle R_i a regular, rectangular mesh is defined. While the mesh width in one coordinate direction is determined by the mesh on the large rectangle, the other can be chosen freely. We triangulate R_i by dividing each of its mesh cells into two triangles. This triangulation induces a triangulation of the inlet as shown in Fig. 2.

3. A preconditioned conjugate gradient method

Let $A \in R^{n \times n}$ be a symmetric, positive semidefinite matrix, and let $\underline{b} \in R^n$ be a vector orthogonal to the kernel of A. Let $\tilde{A} \in R^{n \times n}$ be symmetric and positive semidefinite, with $\ker(\tilde{A}) \subset \ker(A)$. There are several possible preconditioned conjugate gradient algorithms for

$$A\underline{u} = \underline{b} \tag{3.1}$$

using the preconditioner $\tilde{A}$, which are mathematically equivalent but algorithmically different. The version given below is particularly useful for domain decomposition methods. The matrix A only appears in a vector-matrix multiplication with $A-\tilde{A}$. In our application, this is much cheaper than multiplying with A. In addition, we need a procedure which gives us the product of the Moore-Penrose pseudo-inverse $\tilde{A}^+$ of $\tilde{A}$ with an arbitrary vector.

Preconditioned conjugate gradient algorithm:

Let $\underline{u}^{(0)} := \underline{0} \in R^n$.

$\underline{g}^{(0)} := \underline{b}$

Replace $\underline{g}^{(0)}$ by its orthogonal projection onto the range of A.

$$\underline{d}^{(0)} := \underline{g}^{(0)}$$

$$\tilde{\underline{g}}^{(0)} := \tilde{A}^{+}\underline{g}^{(0)}$$

$$\tilde{\underline{d}}^{(0)} := \tilde{\underline{g}}^{(0)}$$

For j=0,1,2,...:

$$\alpha^{(j)} := \left[\underline{g}^{(j)T}\tilde{\underline{g}}^{(j)}\right] \; / \; \left[\tilde{\underline{d}}^{(j)T}((A-\tilde{A})\tilde{\underline{d}}^{(j)}+\underline{d}^{(j)})\right]$$

$$\underline{u}^{(j+1)} := \underline{u}^{(j)}+\alpha^{(j)}\tilde{\underline{d}}^{(j)}$$

$$\underline{g}^{(j+1)} := \underline{g}^{(j)}-\alpha^{(j)}[(A-\tilde{A})\tilde{\underline{d}}^{(j)}+\underline{d}^{(j)}]$$

Replace $\underline{g}^{(j+1)}$ by its orthogonal projection onto the range of A.

$$\tilde{\underline{g}}^{(j+1)} := \tilde{A}^{+}\underline{g}^{(j+1)}$$

$$\beta^{(j)} := \left[\tilde{\underline{g}}^{(j+1)T}\underline{g}^{(j+1)}\right] \; / \; \left[\tilde{\underline{g}}^{(j)T}\underline{g}^{(j)}\right]$$

$$\underline{d}^{(j+1)} := \underline{g}^{(j+1)}+\beta^{(j)}\underline{d}^{(j)}$$

$$\tilde{\underline{d}}^{(j+1)} := \tilde{\underline{g}}^{(j+1)}+\beta^{(j)}\tilde{\underline{d}}^{(j)}.$$

The vectors $\underline{u}^{(j)}$ converge to a solution of (3.1).

The orthogonal projections onto the range of A have no effect in exact arithmetic, but these steps are sometimes necessary to obtain convergence in floating point arithmetic.

4. The domain decomposition algorithm

In this section, we describe the Neumann-Dirichlet domain decomposition method as given in Bjørstad and Widlund (1984,1986). Because of the simple structure of our problem, we can give a simplified convergence analysis. We also present numerical results.

4.1 Notation and preliminary remarks

The finite element approximation of eqs. (1.1), (1.2) can be written as a system of linear equations of the form

$$K\underline{x} = \underline{b}, \tag{4.1}$$

with K, $\underline{x}$ and $\underline{b}$ partitioned as follows:

$$K = \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \underline{b}_3 \end{bmatrix}. \tag{4.2}$$

Here the subscript 1 refers to the nodes of the inlets not belonging to the interfaces between the inlets and the rectangle, the subscript 2 to the nodes in the rectangle not belonging to the interfaces, and the subscript 3 the nodes on the interfaces.

The entries of the stiffness matrix K are integrals of the form

$$\int_{\Omega} \nabla\phi \cdot \nabla\psi \, d\underline{x}, \tag{4.3}$$

where ϕ and ψ are canonical basis functions of the space of piecewise linear finite elements. These integrals can be written in the form

$$\int_{\Omega} = \int_{\Omega^{(1)}} + \int_{\Omega^{(2)}}. \tag{4.4}$$

We obtain a corresponding splitting $K=K^{(1)}+K^{(2)}$, where

$$K^{(1)} = \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & 0 & 0 \\ K_{13}^T & 0 & K_{33}^{(1)} \end{bmatrix}, \tag{4.5}$$

and

$$K^{(2)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & K_{22} & K_{23} \\ 0 & K_{23}^T & K_{33}^{(2)} \end{bmatrix}, \tag{4.6}$$

and thus

$$K_{33} = K_{33}^{(1)} + K_{33}^{(2)}. \tag{4.7}$$

We note that a system of the form

$$\begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix} \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} = \begin{bmatrix} \underline{b}_2 \\ \underline{b}_3 \end{bmatrix} \tag{4.8}$$

is a discrete Neumann problem on $\Omega^{(2)}$.

The following matrix, the so-called Schur complement, plays a central role below:

$$S := K_{33} - K_{13}^T K_{11}^{-1} K_{13} - K_{23}^T K_{22}^{-1} K_{23}. \tag{4.9}$$

Since S can be shown to be a diagonal block in a block Cholesky factorisation of K, it is symmetric and positive semidefinite, and its kernel is span($\underline{1}$), the set of constant vectors. The two subregions each contribute

$$S^{(j)} := K_{33}^{(j)} - K_{j3}^T K_{jj}^{-1} K_{j3}, \quad j=1,2 \tag{4.10}$$

to S. $S^{(1)}$ and $S^{(2)}$ are symmetric and positive semidefinite with $\ker(S^{(j)})$=span($\underline{1}$), and

$$S = S^{(1)} + S^{(2)}. \tag{4.11}$$

The linear system (4.8) is easy to solve, using a standard fast Laplace solvers for rectangular domains. It is easy to see that the solution $\underline{x}_3$ of

$$S^{(2)} \underline{x}_3 = \underline{r}_3 \tag{4.12}$$

is a subvector of the solution of

$$\begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix} \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{r}_3 \end{bmatrix}. \tag{4.13}$$

Therefore eq. (4.12) can be solved with a fast Poisson solver. The application of

$$S^{(1)} = K_{33}^{(1)} - K_{13}^T K_{11}^{-1} K_{13} \tag{4.14}$$

to a vector $\underline{y}_3$ is also easy and inexpensive. The main work required is the solution of the system

$$K_{11} \underline{y}_1 = K_{13} \underline{y}_3. \tag{4.15}$$

We use band Gaussian elimination for this purpose, observing that K_{11} is block diagonal with N_{inlets} blocks. To reduce the band width of a block of K_{11}, we order the unknowns as indicated in Fig. 3. We denote the number of nodes on the interface by m_i.

From the point of view of band width, it does not matter whether the nodes which are closest to the rectangle $\Omega^{(2)}$ are numbered first or last. However, we order them last for the following reason. During the iteration, the right-hand side in (4.15) is nonzero only in the components corresponding to the mesh layer in the inlets closest to $\Omega^{(2)}$, and we only need the same components of $\underline{x}_1$ to form $S^{(1)}\underline{r}_3$. If we order these nodes last, most of the work in the solution phase of the Cholesky algorithm becomes unnecessary. In fact, only the bottom right m_i by m_i submatrices of the triangular factors of

the stiffness matrices for the inlets need to be saved. The amount of work per iteration on the i–th inlet will therefore be proportional to m_i^2, once the stiffness matrices corresponding to the inlets have been factored. Since K_{11} is unaffected by the position of the bottom of $\Omega^{(2)}$, which changes with time, this matrix is factored only once.

4.2 Statement of the algorithm

When solving (4.1), we first reduce the right-hand side $\underline{b}$ to the form

$$\begin{bmatrix} \underline{0} \\ \underline{0} \\ \underline{\rho}_3 \end{bmatrix}. \tag{4.16}$$

This is accomplished by first solving a problem with the matrix

$$\begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix}, \tag{4.17}$$

i.e. a discrete Neumann problem, and then a problem with the matrix K_{11}. A problem of the form

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ \underline{\rho}_3 \end{bmatrix} \tag{4.18}$$

is equivalent to

$$S\underline{y}_3 = \underline{\rho}_3, \tag{4.19}$$

which is solved by using the preconditioned conjugate gradient method, with the preconditioner $S^{(2)}$.

Once the solution $\underline{y}_3$ of (4.19) is known, the complete solution of (4.18) is obtained by solving

$$K_{11}\underline{y}_1 = -K_{13}\underline{y}_3 \tag{4.20}$$

for $\underline{y}_1$, and then

$$\begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix} \begin{bmatrix} \underline{y}_2 \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{\rho}_3 - K_{13}^T\underline{y}_1 - K_{33}^{(1)}\underline{y}_3 \end{bmatrix} \tag{4.21}$$

for $\underline{y}_2$.

This concludes the description of the algorithm.

We note that we never need to compute the elements of S and $S^{(2)}$. The straightforward compu-

tation of S would require the solution of p linear systems with the matrix K_{11}, and the same number of systems with the matrix K_{22}, where p is the total number of nodes on the interfaces between $\Omega^{(1)}$ and $\Omega^{(2)}$. Linear systems with K_{22} are not as easy to solve as those with the matrix (4.17), since the matrix K_{22} represents a discretized boundary value problems with Dirichlet conditions along the interfaces, and Neumann conditions elsewhere. Such a system is, in general, non-separable and fast direct Poisson solvers are not available.

If the elements of $S^{(2)}$ were required, they could also be computed by first computing $(S^{(2)})^{-1}$, by using a fast Poisson solver p times; cf. eqs. (4.12) and (4.13). Thereafter$(S^{(2)})^{-1}$ is inverted. As we have noted, these computations are unnecessary for our algorithm.

4.3 Convergence analysis

From the theory of the conjugate gradient method, it is known that the number of steps needed to reach a prescribed accuracy is bounded above by a function of

$$\kappa = \frac{\max \dfrac{\underline{x}_3^T S \underline{x}_3}{\underline{x}_3^T S^{(2)} \underline{x}_3}}{\min \dfrac{\underline{x}_3^T S \underline{x}_3}{\underline{x}_3^T S^{(2)} \underline{x}_3}}, \tag{4.22}$$

where the maximum and minimum are taken over the orthogonal complement of $\underline{1}$; see e.g. Luenberger (1969), p.296. This function grows linearly with the square root of κ for large κ.

The condition number κ is estimated in two steps; cf. e.g. Bjørstad and Widlund (1986). In the first step, given in Lemma 1, only linear algebra is needed. The second step, given in Lemma 2, is in general more difficult and requires tools from analysis; see Widlund (1987a). However, in the present case, this argument can be very much simplified.

Lemma 1: Let C be the smallest constant such that, for all $\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}$, there is an $\underline{x}_1$ such that

$$\begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} \leq C \cdot \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix} \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}. \tag{4.23}$$

Then

$$\kappa \le C. \tag{4.24}$$

This estimate is not necessarily sharp.

Proof: We first note that the denominator in (4.22) is always ≥1. Therefore

$$\kappa \le \max \frac{\underline{x}_3^T S \underline{x}_3}{\underline{x}_3^T S^{(2)} \underline{x}_3}. \tag{4.25}$$

(4.25) is not necessarily sharp. Consider, e.g., a case where Ω is a rectangle, divided into two halves. Then, by symmetry, $S^{(1)} = S^{(2)}$, and therefore the right-hand side in (4.25) becomes 2, but κ=1.

The right-hand side in (4.25) is bounded by the constant C referred to in Lemma 1. To see this, note that, from the definitions of S and $S^{(2)}$,

$$\max \frac{\underline{x}_3^T S \underline{x}_3}{\underline{x}_3^T S^{(2)} \underline{x}_3} = \max \frac{\begin{bmatrix} -K_{11}^{-1}K_{13}\underline{x}_3 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} -K_{11}^{-1}K_{13}\underline{x}_3 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}}{\begin{bmatrix} -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix} \begin{bmatrix} -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}}. \tag{4.26}$$

The proof of Lemma 1 is completed by showing, by a straightforward computation, that

$$\begin{bmatrix} -K_{11}^{-1}K_{13}\underline{x}_3 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} -K_{11}^{-1}K_{13}\underline{x}_3 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix} \le \begin{bmatrix} \underline{x}_1 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix}^T \begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ -K_{22}^{-1}K_{23}\underline{x}_3 \\ \underline{x}_3 \end{bmatrix} \tag{4.27}$$

for all vectors $\underline{x}_1$ and $\underline{x}_3$. □

We note that the right-hand side in (4.25) is exactly equal to the constant C.

For Lemma 2, the following notation is needed. I_i is the i–th inlet, and R_i the corresponding rectangle; cf. Section 2 and Fig. 2. Recall that the mesh on I_i is the image of a regular rectangular mesh on R_i. Let h_i be the mesh width in R_i in the direction normal to the interface.

Lemma 2: Assume that the mesh ratios $\frac{h_x}{h_i}$ and $\frac{h_y}{h_i}$ are uniformly bounded. Then the constant C of Lemma 1 is uniformly bounded.

Proof: Let

$$\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} \tag{4.28}$$

be a vector representing a finite element function ϕ on $\Omega^{(2)}$. The quadratic form on the right-hand side of (4.23) is the Dirichlet integral of ϕ. Similarly, the left-hand side of (4.23) is the Dirichlet integral of a finite element function which extends ϕ to all of Ω. To estimate C, we should therefore study the extension of finite element functions from $\Omega^{(2)}$ to Ω. A general extension theorem of this kind for conforming finite elements has been established by Widlund (1987a).

When studying the extension problem for the current geometry, we first make three simplifying assumptions: 1) The inlets are rectangular. 2) The mesh widths in the inlets are the same as those of $\Omega^{(2)}$. 3) The mirror images of the inlets are completely contained in $\Omega^{(2)}$ and do not overlap each other; see Fig. 4.

The finite element function ϕ is extended into the inlets as an even function with respect to the direction normal to the interface. The Dirichlet integral over $\Omega^{(1)}$ of the extended function is bounded by the Dirichlet integral of ϕ over $\Omega^{(2)}$, and thus

$$C \leq 2. \tag{4.29}$$

We note that this argument fails if (1.2) were a Dirichlet condition, since the extended function would have to vanish on the three outer sides of each inlet.

The remainder of the proof consists of the step-by-step removal of the assumptions 1-3. We begin with assumption 3. If the reflected images of the inlets in $\Omega^{(2)}$ overlap each other, (4.29) is replaced by

$$C \leq 4, \tag{4.30}$$

since no point in $\Omega^{(2)}$ can be covered by more than three reflected images.

If one of the inlets is so long that its reflected image is not entirely contained in $\Omega^{(2)}$, the

reflection does not define the extension in the entire inlet, but only in a piece of it. However, the extension can be completed by repeated reflection. Let, for example, the left side of $\Omega^{(2)}$ be the line $x=0$, and the right side of $\Omega^{(2)}$ be the line $x=a$. If (x,y) is a point in a left inlet, with $-2a<x<-a$, then

$$\bar{\phi}(x,y) := \bar{\phi}(-2a-x,y), \tag{4.31}$$

and

$$\bar{\phi}(-2a-x,y) := \phi(2a+x,y). \tag{4.32}$$

This again leads to a larger, but mesh-independent C.

Suppose now that assumption 2 is violated, e.g. that the horizontal mesh width h in a left inlet differs from the horizontal mesh width h_x in $\Omega^{(2)}$. The extension function is given by

$$\bar{\phi}(-x,y) := \phi(x\cdot\frac{h_x}{h},y). \tag{4.33}$$

The Dirichlet integral of $\bar{\phi}$ can easily be estimated in terms of the Dirichlet integral of ϕ. If $h_x \le h_i$ for all i, the previous estimate for C remains unchanged. Otherwise, it is enlarged by a factor q, where q is the maximum of all quotients $\frac{h_x}{h_i}$ and $\frac{h_y}{h_i}$. This factor is, by the assumption of Lemma 2, bounded independently of the mesh.

Assumption 1 can be eliminated by using a similar argument. The inlets are images of rectangular regions, and the corresponding mapping functions ψ_i naturally enter. The estimate for C is then enlarged by the factor

$$\max_i \max_{\underline{x}} \rho[\det(D\psi_i(\underline{x}))(D\psi_i(\underline{x}))^{-T}(D\psi_i(\underline{x}))^{-1}] + o(1), \tag{4.34}$$

where $\rho[\cdots]$ denotes the spectral radius, and ψ_i are the bilinear mappings used in the mesh construction, see Section 2. This follows from an estimate of the Dirichlet integral and a change of variables. □

From Lemmas 1 and 2, we obtain the convergence theorem:

Theorem: Let the mesh ratios be bounded as in Lemma 2. Then the condition number κ defined in (4.22) is bounded independently of the mesh.

This analysis and the numerical results below show that $S^{(2)}$ is an excellent preconditioner for S.

It is also possible to use $S^{(1)}$ as a preconditioner for S. As we noticed previously, there is a considerable difference between a Neumann and a Dirichlet problem. For a Dirichlet problem, $S^{(1)}$ would be the better choice; see e.g. Fig. 8 in Anderson (1985). The reason is that in the Dirichlet case, the extension of certain finite element functions from $\Omega^{(2)}$ into $\Omega^{(1)}$ will introduce large gradients, since we have to impose homogeneous Dirichlet conditions on the outer sides of each inlet. This results in a large Dirichlet integral over $\Omega^{(1)}$. On the other hand, the extension from $\Omega^{(1)}$ to $\Omega^{(2)}$ can easily be accomplished in the Dirichlet case if assumptions 1 to 3 are satisfied. We use reflections and then extend the resulting function by zero to the rest of $\Omega^{(2)}$, obtaining an extended function with twice as big a Dirichlet integral.

For Neumann problems, we are not able to establish that preconditioning with $S^{(1)}$ leads to a large or small condition number. From a practical point of view, the use of $S^{(2)}$ is clearly preferable, since using $S^{(1)}$ as a preconditioner would make it necessary to solve non-separable problems on $\Omega^{(2)}$.

4.4 Numerical results

We have conducted experiments on the regions shown in Figs. 1 and 4. The size of the mesh, including the mesh points on the boundary, is 33μ by 33μ in the rectangle $\Omega^{(2)}$, and 3μ by 9μ in each of the inlets, with μ=1,2,3 or 4. Tables I and II give information on the rate of convergence of the iterative method. The choice of boundary data has no significant influence on the performance. In these runs, random boundary data are used.

We give the number of iterations required for

$$|\underline{\rho}_3 - S\underline{y}_3|_2 \le \varepsilon |\underline{\rho}_3|_2, \tag{4.35}$$

cf. eq. (4.19), and also determine the condition number κ by using the coefficients generated by the conjugate gradient method; see e.g. O'Leary and Widlund (1979).

5. Using an inexact solver on the rectangle

5.1 A different view of the domain decomposition algorithm

The iteration described in Section 4 requires an application of $(S^{(2)})^+$, and thus a call to a fast Laplace solver on $\Omega^{(2)}$ in every step. It is not practical to replace this solver by an approximate solver without making other changes in the algorithm, for the following reason. The use of an inexact solver can be interpreted as replacing $(S^{(2)})^+$ by an approximation. It is easy to construct an inexact solver such that the approximation takes the form $(\tilde{S}^{(2)})^+$, with a symmetric, positive semidefinite matrix $\tilde{S}^{(2)}$ with $\ker(\tilde{S}^{(2)}) = \mathrm{span}(\underline{1})$. However, the resulting domain decomposition method requires, in each step, one application of $S-\tilde{S}^{(2)}$ and one application of $(\tilde{S}^{(2)})^+$, or alternatively, one application of S and one of $(\tilde{S}^{(2)})^+$. This is not practical because applications of $S-\tilde{S}^{(2)}$ or S would require the exact solution of linear systems with the matrix K_{22}.

The following modification of the algorithm of Section 4.2 makes the use of an inexact solver possible. We first reduce the right-hand side of (4.1) to the form

$$\begin{bmatrix} \underline{0} \\ \underline{\rho}_2 \\ \underline{\rho}_3 \end{bmatrix}. \tag{5.1}$$

This is accomplished by solving a problem with the matrix K_{11}, resulting in a problem of the form

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & K_{22} & K_{23} \\ K_{13}^T & K_{23}^T & K_{33} \end{bmatrix} \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{\rho}_2 \\ \underline{\rho}_3 \end{bmatrix}. \tag{5.2}$$

This system can be reduced to

$$R \begin{bmatrix} \underline{y}_2 \\ \underline{y}_3 \end{bmatrix} = \begin{bmatrix} \underline{\rho}_2 \\ \underline{\rho}_3 \end{bmatrix}, \tag{5.3}$$

with

$$R := \begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33} - K_{13}^T K_{11}^{-1} K_{13} \end{bmatrix}. \tag{5.4}$$

Eq. (5.3) is solved by using the preconditioned conjugate gradient method, with the preconditioner

$$\tilde{R} := \begin{bmatrix} K_{22} & K_{23} \\ K_{23}^T & K_{33}^{(2)} \end{bmatrix}. \tag{5.5}$$

Once the solution of (5.3) is known, the complete solution of (5.2) is obtained by solving

$$K_{11}\underline{y}_1 = -K_{13}\underline{y}_3. \tag{5.6}$$

This concludes the description of the modified algorithm.

By replacing Lemma 1 by Lemma 3, the convergence analysis for the modified algorithm can be completed.

Lemma 3: The bound C of Lemma 1 is also a bound on

$$\max \frac{\begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}^T R \begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}}{\begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}^T \tilde{R} \begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}} \Bigg/ \min \frac{\begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}^T R \begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}}{\begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}^T \tilde{R} \begin{bmatrix}\underline{x}_2\\ \underline{x}_3\end{bmatrix}}. \tag{5.7}$$

For the design of algorithms using inexact solvers on both subregions, see Bramble, Pasciak and Schatz (1986), Widlund (1987b,c).

5.2 Multigrid cycles with symmetric iteration matrices

In Section 5.3, the fast solver on $\Omega^{(2)}$ will be replaced by a multigrid cycle. We will require the iteration matrix of the multigrid cycle to be symmetric. This can be accomplished as follows.

Consider a system of n linear equations

$$A\underline{x} = \underline{b} \tag{5.8}$$

with $A=A^T\geq 0$. We first assume that A is invertible, and discuss the case of a non-trivial kernel later.

We first recall the definition of the well-known two-level correction cycle; see e.g. Stüben and Trottenberg (1982). Let m be an integer; m is usually smaller than the number n of equations in the system to be solved. Let $B\in R^{m\times m}$ be a symmetric matrix, which should be chosen as an m–dimensional approximation to A^{-1}. Let R be a linear mapping $R^n\rightarrow R^m$. A two-level correction step is defined by

$$\underline{x}_{new} := \underline{x}_{old} + R^T B R(\underline{b} - A\underline{x}_{old}) \tag{5.9}$$

The complete two-level cycle is obtained by supplementing (5.9) with a relaxation method. We consider Gauss-Seidel block relaxation here. Let I be any subset of $\{1,\cdots,n\}$. Given an approximation $\underline{x}$ for $A^{-1}\underline{b}$, one can determine $\underline{\delta}\in R^n$ with $\delta_i=0$ for $i\notin I$, and such that the j-th component of $\underline{b}-A(\underline{x}+\underline{\delta})$ is zero for all $j\in I$. We then say that $\underline{x}+\underline{\delta}$ is obtained from $\underline{x}$ by the relaxation of the block I. Let $I_1, I_2,\cdots, I_r$ be subsets of $\{1,\cdots,n\}$, the union of which is $\{1,\cdots,n\}$.

Two-level correction cycle:

Choose $\underline{x}_0$ and generate $\underline{x}_1,\underline{x}_2,\cdots$ as follows.

$\underline{x}_{j+1}$ is obtained from $\underline{x}_j$ by relaxing over the blocks $I_1,I_2,\cdots,I_r$, carrying out one step of (5.9), and relaxing over the blocks $I_r,I_{r-1},\cdots,I_1$.

The iteration (5.9) and each individual block relaxation step have symmetric iteration matrices with respect to the energy product, i.e. the inner product defined by A. We can therefore conclude:

Lemma 4: The iteration matrix M of the two-level correction cycle is symmetric with respect to the energy product.

A mapping $\tilde{G}$ is defined as follows. For $\underline{b}\in R^n$, $\tilde{G}\underline{b}$ is the result of a two-level correction cycle for the problem

$$A\underline{x} = \underline{b}, \tag{5.10}$$

starting with the initial guess $\underline{x}=\underline{0}$. Denoting

$$G := A^{-1}, \tag{5.11}$$

we obtain

$$\tilde{G} = (I-M)G. \tag{5.12}$$

Lemma 5: If M is symmetric in the energy inner product, and if $\rho(M)<1$, then $\tilde{G}$ is symmetric in the euclidean inner product and positive definite, and

$$\mathrm{cond}(\tilde{G}^{-1/2}G\tilde{G}^{-1/2}) \le \frac{1+\rho(M)}{1-\rho(M)}. \tag{5.13}$$

Proof: Since M is symmetric in the energy product, MG and thus $\tilde{G}$ are symmetric in the euclidean inner product. Next we show that $\tilde{G}$ is positive definite if $\rho(M)<1$. $\tilde{G}$ is congruent with

$G^{-1/2}(I-M)G^{1/2}$, which is similar to $I-M$. Since M is symmetric in the energy inner product, M has only real eigenvalues, and if $\rho(M)<1$, $I-M$ has only positive real eigenvalues. Using Sylvester's theorem, it follows that $\tilde{G}$ has only positive eigenvalues. We now estimate $\mathrm{cond}(\tilde{G}^{-1/2}G\tilde{G}^{-1/2})$:

$$\mathrm{cond}(\tilde{G}^{-1/2}G\tilde{G}^{-1/2}) = \mathrm{cond}(G^{-1/2}\tilde{G}G^{-1/2}) = \mathrm{cond}(G^{-1/2}(I-M)G^{1/2}) \tag{5.14}$$

$$= \frac{\lambda_{\max}(I-M)}{\lambda_{\min}(I-M)} = \frac{1-\lambda_{\min}(M)}{1-\lambda_{\max}(M)} \le \frac{1+\rho(M)}{1-\rho(M)}. \tag{5.15}$$

This concludes the proof of Lemma 5. □

Since we are interested in multigrid cycles for Neumann problems, we will now study the case of a constant null vector. We consider

$$A\underline{x} = \underline{b} \tag{5.16}$$

and assume that $\underline{b}\in R^n$ satisfies the compatibility condition $\underline{b}\cdot\underline{1}=0$. (5.9) remains unchanged. If I is a proper subset of $\{1;\cdots;n\}$, then the corresponding principal minor of A is invertible and therefore the relaxation of the block I is well-defined. The definition of the two-level cycle remains unchanged. The solution is defined only modulo constants, and therefore the iteration matrix M is defined as a mapping of the orthogonal complement of span($\underline{1}$) into itself.

For $\underline{b}\in R^n$ with $\underline{b}\cdot\underline{1} = 0$, let $\underline{x}^{(1)}$ be the result of a two-level cycle for $A\underline{x}=\underline{b}$, with the initial guess $\underline{x}^{(0)}=\underline{0}$, and let $\tilde{G}b$ be the projection of $\underline{x}^{(1)}$ onto the orthogonal complement of span($\underline{1}$).

Lemma 6: $\tilde{G}$ is symmetric with respect to the euclidean inner product, and a positive definite mapping of the orthogonal complement of span($\underline{1}$) onto itself. Denoting this restriction by $\tilde{G}^*$, we have:

$$\mathrm{cond}\left[(\tilde{G}^*)^{-1/2}G^*(\tilde{G}^*)^{-1/2}\right] \le \frac{1+\rho(M)}{1-\rho(M)}, \tag{5.17}$$

where G^* is the restriction of G to the orthogonal complement of span($\underline{1}$).

The proof is similar to that of Lemma 5.

The discussion of this section has been restricted to two-level cycles. However, it is easy to obtain analogous statements for arbitrarily many levels, using, e.g., V- or W-cycles; see e.g. Stüben and Trottenberg (1982) for the definition of these cycles.

5.3 The domain decomposition algorithm using a multigrid cycle as an inexact solver

We have carried out numerical experiments with the modified domain decomposition algorithm described in Section 5.1, where the exact fast solver is replaced by a multigrid V-cycle. The ratio of the mesh widths of consecutive levels is $\frac{1}{2}$. Piecewise bilinear interpolation is used to transfer the corrections from a given level to the next finer level, and residuals are transferred from a given level to the next coarser level by using the adjoint of the bilinear interpolation operator. The relaxation method is red-black Gauss-Seidel iteration. Three half sweeps (red-black-red) are carried out before and after each coarse grid correction step.

We find numerically that the spectral radius of the iteration matrix of this cycle is not larger than 0.18 on $n+1$ by $n+1$ meshes, where n is a power of 2. By rearranging the relaxation sweeps in a non-symmetric way, we could obtain smaller multigrid convergence factors for the same amount of work. We have, however, used the symmetric arrangement since it is needed for the analysis of the method.

By using the framework of Sections 5.1 and 5.2, we can interpret this method as a preconditioned conjugate gradient method for eq. (5.3), using a symmetric, positive definite preconditioner $\tilde{R}^{\bullet}$ with $\ker(\tilde{R}^{\bullet}) = \mathrm{span}(\underline{1})$, and with

$$\max \frac{\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T R \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}}{\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T \tilde{R}^{\bullet} \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}} \Big/ \min \frac{\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T R \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}}{\begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}^T \tilde{R}^{\bullet} \begin{bmatrix} \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}} \leq \frac{1+0.18}{1-0.18} C \approx 1.44C. \tag{5.18}$$

Here C is the constant defined in Lemma 1.

Table III repeats the tests of Table 1, using the method described above. The numbers of iterations in Table III are somewhat larger than those in Table I, but the method is substantially faster because the multigrid cycle is less expensive than a direct solver.

6. Further directions

In this section, we will discuss the possibility of using faster solvers on the rectangle and mesh refinement in neighborhoods of the re-entrant corners.

We begin by noticing that during the iteration, the solution of the discrete elliptic problem on the rectangle is only required on and next to the interfaces, i.e. at relatively few points. Similarly, the right hand side of the equation differs from zero only on the interfaces. Problems of this kind were considered by Banegas (1978), and more recently by several Soviet scientists; cf. e.g. Bakhvalov and Orekhov (1982), Kuznetsov and Matsokin (1982). When solving the discrete Poisson problem on an n by n mesh using Bakhvalov's method, only on the order of $n\log(n)^2$ operations are required to obtain the solution on one mesh line within a prescribed accuracy. When using this method, the cost of solving our problem on the entire region to prescribed accuracy would asymptotically, i.e. for sufficiently fine meshes, be dominated by the cost of the final step in which the solution is obtained at all mesh points.

The problem of the lack of accuracy caused by the re-entrant corners could be addressed systematically, using tools similar to those of this paper. In the work of Bai and Brandt (1987), Bramble, Ewing, Pasciak and Schatz (1987), Hart and McCormick (1987), McCormick (1984), and McCormick and Thomas (1986), the questions of designing accurate approximations on locally refined meshes and the effective solution of the resulting linear systems of equations are addressed. Some of these algorithms are multigrid methods, while others require exact solvers for the unrefined problems as well as the refined models on the patches of refinement. It has been demonstrated that certain of these methods work quite well for a considerable number of refinement levels. We note that the relationship between domain decomposition and iterative refinement methods will be explored in a forthcoming report; see Widlund (1987b).

What comes to mind for our current problem, would be to design a refinement of the mesh locally around the re-entrant corners, possibly using several successive refinements. If the refined patches of the rectangle were logically rectangular, then iterative methods could be designed using only fast Poisson solvers, as in the algorithm of Section 4. We might prefer to handle the inlets by

Gaussian elimination, while the finite element problems on the locally refined meshes of $\Omega^{(2)}$ could be solved inexactly, e.g. by one step of an iterative refinement algorithm; cf. Hart and McCormick (1987) and Widlund (1987c). A suitable domain decomposition algorithm would be used as an outer iteration. Such an algorithm would be similar to the one outlined in Section 5. It is also possible to use domain decomposition on the different mesh levels separately and an iterative refinement method as an outer iteration.

Acknowledgements

Much of this work was done while the second author was enjoying the hospitality of Alexandre Chorin and the Department of Mathematics of the University of California at Berkeley. The authors also wish to thank Gene Golub for his interest and helpful comments, and James Sethian for proposing the problem.

References

C. R. Anderson, On domain decomposition, Manuscript CLaSSiC-85-09, Center for Large Scale Scientific Computation, Stanford University (1985).

D. Bai and A. Brandt, Local mesh refinement multilevel techniques, SIAM J. Sci. Stat. Comput. 8, No. 2, pp. 109-134 (1987).

N.S. Bakhvalov and M.Yu. Orekhov, On fast methods of solving Poisson's equations, U.S.S.R. Comput. Maths. Math. Phys. 22, No. 6, pp. 107-114 (1982).

A. Banegas, Fast Poisson solvers for problems with sparsity, Math. Comp. 32, No. 142, pp. 441-446 (1978).

P. E. Bjørstad and O. B. Widlund, Solving elliptic problems on regions partitioned into substructures, *in* "Elliptic Problem Solvers II", G. Birkhoff and A. Schoenstadt, eds., Academic Press, pp. 245-256 (1984).

P. E. Bjørstad and O. B. Widlund, Iterative methods for the solution of elliptic problems on regions partitioned into substructures, SIAM J. Numer. Anal. 23, pp. 1097-1120 (1986).

J. H. Bramble, R. E. Ewing, J. E. Pasciak and A. H. Schatz, A preconditioning technique for the efficient solution of problems with local grid refinement, Computer Methods in Applied Mechanics and Engineering , to appear (1987).

J. H. Bramble, J. E. Pasciak and A. H. Schatz, An iterative method for elliptic problems on regions partitioned into substructures, Math. Comp. 46, No. 173, pp. 361-369 (1986).

R. Glowinski and J. Périaux, "Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations", held in Paris, France, January 7-9, 1987, SIAM, Philadelphia (1987).

P. Grisvard, "Elliptic Problems in Nonsmooth Domains", Pitman, Boston (1985)

L. Hart and S. McCormick, Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: Computational Analysis, Technical Report, Computational Mathematics Group, University of Colorado at Denver (1987).

Yu. A. Kuznetsov and A. M. Matsokin, Partial solution of systems of linear algebraic equations, *in* "Numerical Methods in Applied Mathematics", Nauka, Sibirsk. Otdel., Novosibirsk, pp. 143-163 (1982) (in Russian).

D. G. Luenberger, "Optimization by Vector Space Methods", John Wiley and Sons, New York (1969).

D. P. O'Leary and O. B. Widlund, Capacitance matrix methods for the Helmholtz equation on general three-dimensional regions, Math. Comp. 33, No. 147, pp. 849-879 (1979).

S. McCormick, Fast adaptive composite grid (FAC) methods: Theory for the variational case, Computing, Suppl. 5, pp. 115-121 (1984).

S. McCormick and J. Thomas, The fast adaptive composite grid (FAC) method for elliptic equations, Math. Comp. 46, No. 174, pp. 439-456 (1986).

J. A. Sethian, Turbulent combustion in open and closed vessels, J. Comp. Phys. 54, pp. 425-456 (1984).

J. A. Sethian, Development of large eddies in moving pistons, to be submitted to J. Comp. Phys. (1987).

J. A. Sethian and A. F. Ghoniem, Validation of vortex methods, to appear in J. Comp. Phys.; also LBL-Report 21767 (1986).

K. Stüben and U. Trottenberg, Multigrid methods: fundamental algorithms, model problem analysis and applications, *in* "Lecture Notes in Mathematics", vol. 960, Springer-Verlag, Berlin (1982).

O. B. Widlund, An extension theorem for finite element spaces and three applications, *in* "Proceedings of the Second GAMM-Seminar, Kiel, January 17 to 19, 1986", Notes on Numerical Fluid Mechanics 16, pp. 110-122 (1987a).

O. B. Widlund, Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane, to appear in Glowinski and Périaux (1987); also Technical Report 287, Computer Science Department, Courant Institute (1987b).

O. B. Widlund, A Comparison of Some Domain Decomposition and Iterative Refinement Algorithms for Elliptic Finite Element Problems, to appear as a Technical Report, Computer Science Department, Courant Institute (1987c).

Table I. Experiments for the region in Fig. 1. The size of the mesh, including the mesh points on the boundary, is 33μ by 33μ in the rectangle $\Omega^{(2)}$, and 3μ by 9μ in each of the inlets.

μ	condition number	ε	number of iterations
1	2.88	10^{-1}	3
		10^{-2}	4
		10^{-3}	5
		10^{-4}	5
		10^{-5}	6
		10^{-6}	7
		10^{-7}	7
		10^{-8}	8
		10^{-9}	9
		10^{-10}	9
2	3.07	10^{-1}	3
		10^{-2}	4
		10^{-3}	6
		10^{-4}	7
		10^{-5}	8
		10^{-6}	8
		10^{-7}	9
		10^{-8}	10
		10^{-9}	11
		10^{-10}	13

μ	condition number	ε	number of iterations
3	3.12	10^{-1}	3
		10^{-2}	4
		10^{-3}	6
		10^{-4}	7
		10^{-5}	9
		10^{-6}	10
		10^{-7}	10
		10^{-8}	11
		10^{-9}	12
		10^{-10}	13
4	3.14	10^{-1}	3
		10^{-2}	4
		10^{-3}	6
		10^{-4}	7
		10^{-5}	9
		10^{-6}	10
		10^{-7}	11
		10^{-8}	12
		10^{-9}	13
		10^{-10}	14

Table II. Experiments for the region in Fig. 4. The size of the mesh, including the mesh points on the boundary, is 33μ by 33μ in the rectangle $\Omega^{(2)}$, and 3μ by 9μ in each of the inlets.

μ	condition number	ε	number of iterations
1	1.90	10^{-1}	2
		10^{-2}	3
		10^{-3}	4
		10^{-4}	4
		10^{-5}	5
		10^{-6}	5
		10^{-7}	5
		10^{-8}	6
		10^{-9}	7
		10^{-10}	8
2	2.00	10^{-1}	2
		10^{-2}	3
		10^{-3}	4
		10^{-4}	5
		10^{-5}	6
		10^{-6}	6
		10^{-7}	7
		10^{-8}	8
		10^{-9}	8
		10^{-10}	9

μ	condition number	ε	number of iterations
3	2.00	10^{-1}	2
		10^{-2}	3
		10^{-3}	4
		10^{-4}	5
		10^{-5}	6
		10^{-6}	7
		10^{-7}	7
		10^{-8}	8
		10^{-9}	9
		10^{-10}	9
4	2.00	10^{-1}	2
		10^{-2}	3
		10^{-3}	4
		10^{-4}	5
		10^{-5}	6
		10^{-6}	7
		10^{-7}	7
		10^{-8}	8
		10^{-9}	9
		10^{-10}	10

Table III. Experiments for the region in Fig. 1, using the method of Section 5. The size of the mesh, including the mesh points on the boundary, is 33μ by 33μ in the rectangle $\Omega^{(2)}$, and 3μ by 9μ in each of the inlets.

μ	condition number	ϵ	number of iterations
1	3.51	10^{-1}	3
		10^{-2}	4
		10^{-3}	6
		10^{-4}	8
		10^{-5}	9
		10^{-6}	10
		10^{-7}	11
		10^{-8}	12
		10^{-9}	15
		10^{-10}	15
2	3.75	10^{-1}	3
		10^{-2}	5
		10^{-3}	7
		10^{-4}	8
		10^{-5}	10
		10^{-6}	11
		10^{-7}	13
		10^{-8}	15
		10^{-9}	17
		10^{-10}	19

μ	condition number	ε	number of iterations
4	3.84	10^{-1}	3
		10^{-2}	5
		10^{-3}	7
		10^{-4}	9
		10^{-5}	11
		10^{-6}	13
		10^{-7}	15
		10^{-8}	16
		10^{-9}	18
		10^{-10}	19

List of legends for the figures

Figure 1: The region $\Omega = \Omega^{(1)} \cup \Omega^{(2)}$. $\Omega^{(1)}$ is the union of the small pieces, and $\Omega^{(2)}$ is the rectangle.

Figure 2: Solid: I_i and its triangulation. Dashed: R_i and its triangulation.

Figure 3: Ordering of the unknowns in the inlets.

Figure 4: A domain satisfying assumptions 1 and 3 in the proof of Lemma 2.

FIGURE 1

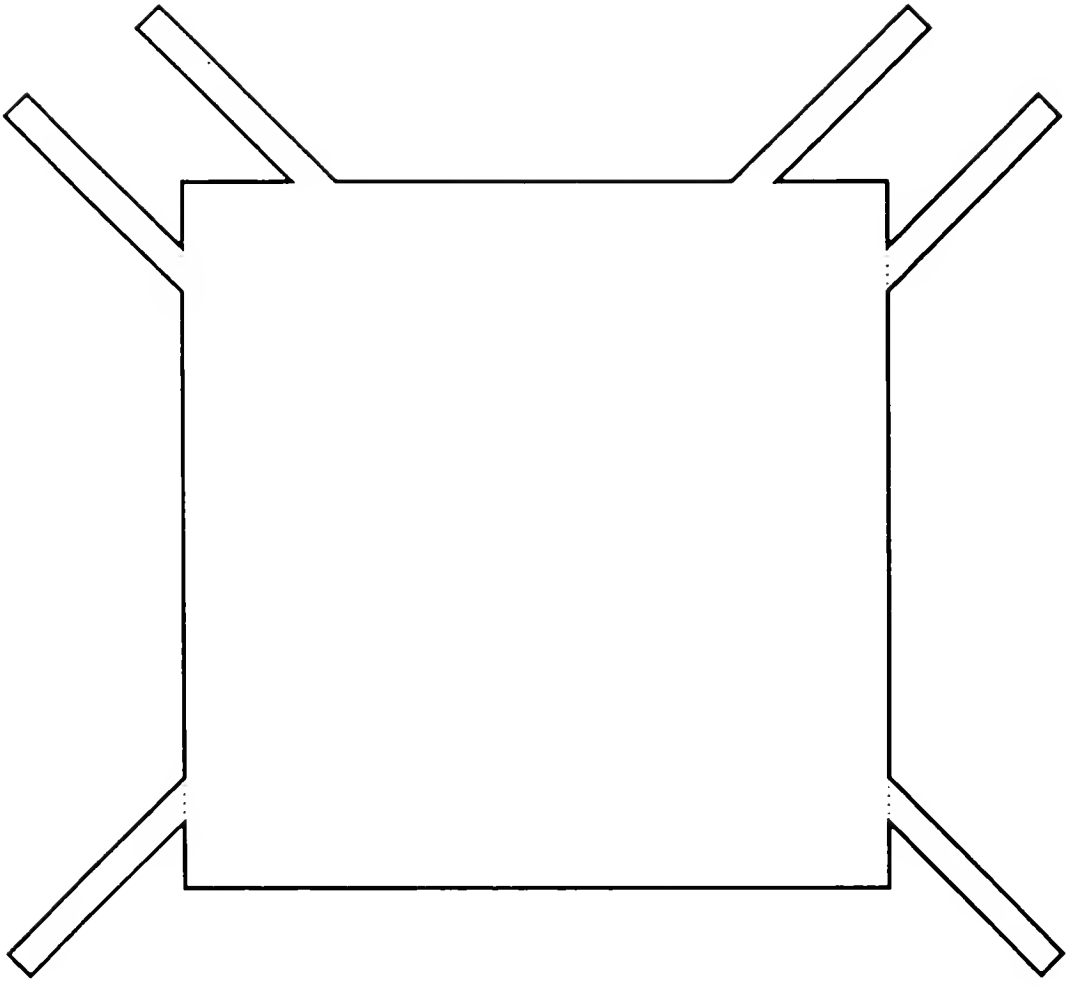


FIGURE 2

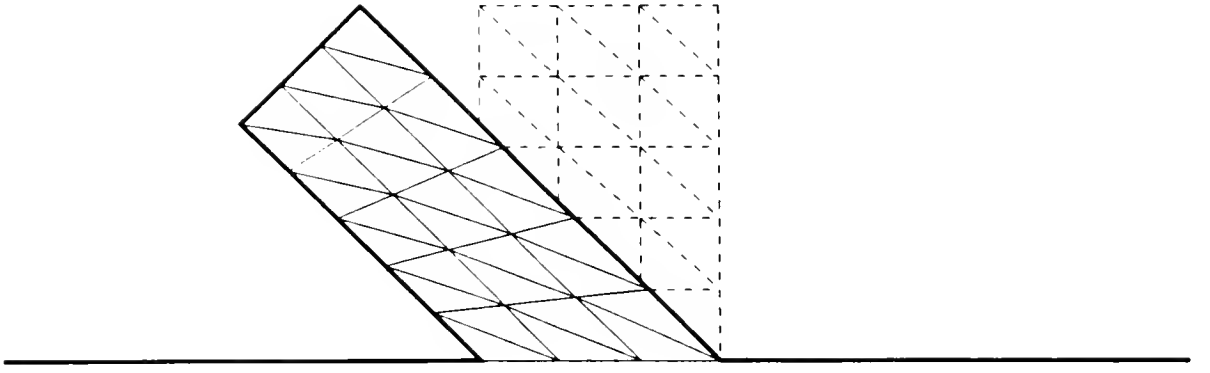


FIGURE 3

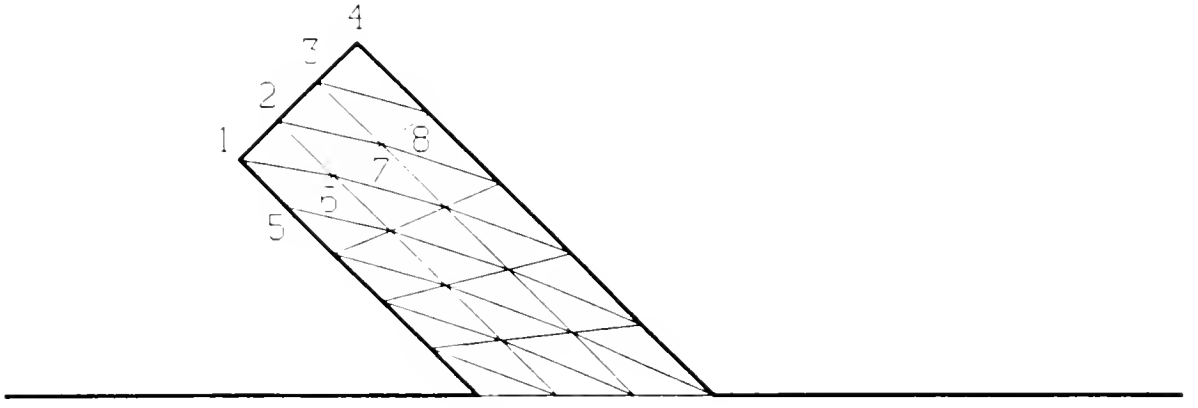
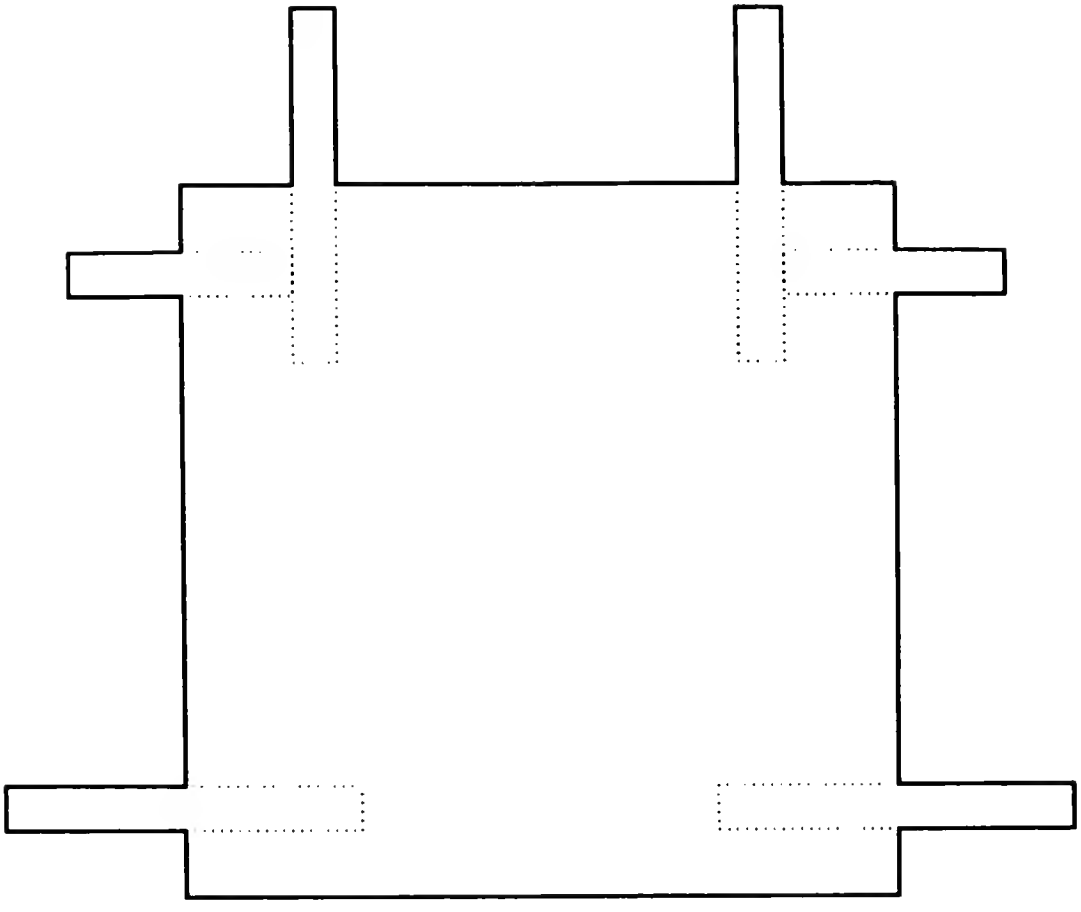


FIGURE 4



NYU COMPSCI TR-315 c.1
Borgers, Christoph
A domain decomposition
Laplace solver for
internal combustion engine